Abstract

In this paper, the Bayesian Data Reduction Algorithm (BDRA) is extended to classify discrete test observations given the training data contains feature vectors which are missing values. Two methods are used to model missing features in the BDRA, where performance is compared to a neural network using both simulated and real data. In general, it is shown that the BDRA is superior to the neural network.

1 Introduction

In this paper, the Bayesian Data Reduction Algorithm (BDRA) is modified to classify test observations given that the training data is missing feature information. Notice, by missing features it is meant that each of the $N_k$ feature vectors of the training data for the $k$th class are assumed to be made up of either or both of the following two observation types: features which are represented by discrete values, and missing features which have no values. For example, with three binary features a possible feature vector that is missing a single feature might appear as $(1, 1, x)$ where $x$, having a cardinality of two, represents the missing value. Notice, the missing features are assumed to appear according to an unknown probability distribution, and in simulated training data a uniform random variable is used to control their frequency of occurrence.

Now, in the BDRA the missing feature information is modeled using two different methods. In Method 1 the Dirichlet prior is extended to accommodate missing features in the natural way. That is, each missing feature is assumed to be uniformly distributed over its range of possible values. In Method 2 the number of discrete levels for each feature is increased by one so that all missing values for that feature are assigned to the same discrete level. To illustrate performance, both methods are compared to a neural network using simulated and real data.

2 The BDRA With Missing Features

The BDRA is formulated using the empirical probability of error, and for one test observation (i.e., $N_y = 1$) it is given by [1]

$$P(e | W) = \sum_y \sum_y P(H_k) I_{(x \leq x_t)} f(y|w_x, H_k) + P(H_i) I_{(x > x_t)} f(y|w_x, H_i)$$

where, based on Method 1, $z_k = f(y_1 = 1|x_k, H_k) = (N_k + M - 1)! \left( \sum_{i \in S_1} \frac{1}{w_{k,i}} \right)^{N_k + M - 1} \left( \sum_{i \in S_2} \frac{1}{w_{k,i}} \right)^{N_k}$, $k \in \{\text{class 1, class 2}\}$, and $k \neq i$; $P_k$ is the number of discrete symbols; $w_{k,i}$ is a single observation of a feature vector in the training data, and $| w_{k,i} |$ is its cardinality; $w_{k,i}$ is a single observation of a feature vector in the test data, and $| w_{k,i} |$ is its cardinality; $S_i$ is defined as all those $w_{k,i}$ that can take on symbol $i$; $S_{y,i}$ is defined as all those $w_{k,i}$ that can take on symbol $i$; $X \equiv (x_k, x_i)$ is all training data; $z_{k,i} = \sum_{i \in S_1} \frac{1}{w_{k,i}}$ is the number of the $i^{th}$ symbol in the training data for class $k$ (appearing with probability $p_{k,i}$), and $N_k \left\{ N_k = \sum_{i=1}^M x_{k,i} \right\}$; $y_i = \sum_{i \in S_{y,i}} \frac{1}{w_{k,i}}$ is the number of the $i^{th}$ symbol in the test data (appearing with probability $p_{y,i}$), and $N_y \left\{ N_y = \sum_{i=1}^M y_i \right\}$; $I_{(x_t)}$ is the indicator function.

In Method 2 the number of discrete levels is increased by one for each feature that has missing values (this is also the "filling in" type method used for the neural network). Thus, the $S_i$ of formula (1) all equal unity

1Supported by a Naval Undersea Warfare Center In-House Laboratory Independent Research grant.
2Supported by ONR under contract N00014-97-1-0502, and by AFOSR under contract F49620-97-1-0198.
3The BDRA is based on the Combined Bayes Test (CBT), which uses the Dirichlet distribution as a noninformative prior [1].
and the initial quantization complexity, $M$, is increased according to the additional levels of the missing features (e.g., six binary features with two of them missing values increases the initial quantization complexity from $M = 64$ to $M = 144$).

For either Method 1 or Method 2 the BDRA is implemented by using the following iterative steps: 1) Using the initial training data formula (1) is used to compute $P(e | X; M)$; 2) Begin with an arbitrarily selected feature and sum (i.e., merge) the training data of those symbols that correspond to reducing it by one quantization level; 3) Use the newly merged training data, $X'$, and the new quantization, $M'$, and again compute $P(e | X'; M')$; 4) Repeat items two and three for all adjacent feature quantizing levels, and all remaining features; From item four select the minimum of all computed $P(e | X'; M')$, and choose this as the new training data configuration; 5) Repeat items two through five until the probability of error decreases no further, or until $M' = 2$.

### Table 1: Performance With the ACCD

<table>
<thead>
<tr>
<th></th>
<th>Method 1</th>
<th>Method 2</th>
<th>Neural Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(e)$</td>
<td>0.165</td>
<td>0.135</td>
<td>0.284</td>
</tr>
</tbody>
</table>

Figure 1 shows error probabilities for the BDRA (using Method 1), a neural network (trained and tested using the Neural Network Toolbox of Matlab), and the optimal (clairvoyant) test as a function of the number of relevant features for each class. The term “relevant” means that those features are distributed uniquely for each class with the remaining ones distributed the same amongst the classes. Additionally, there are twenty five samples of training data for each class, and results are based on an average of one hundred independent trials. For the case shown in Figure 1, a 0.25 probability exists for each class that up to three randomly selected features will be missing from the feature vectors in the training set. Additionally, the neural network is trained on ternary valued feature vectors where a third discrete level is used for each missing feature. Observe that the BDRA is superior to the neural network by achieving an overall lower probability of error (except when all features are relevant). With this, and not shown here, Method 2 was also applied and found to perform similar to Method 1.

In the next application the BDRA is trained and tested on the Australian Credit Card Data (ACCD) (for more on this data see the references in [1]). The ACCD contains fifteen features (six continuous and nine discrete), where five percent of the feature vectors have one or more missing values. Notice, the continuous valued features were discretized to binary values using percentiles (a finer quantization for these features produced worse results). Also, in applying the BDRA to the ACCD the data were randomly partitioned into training (518 samples) and test (172 samples) sets. With this, all results shown are based on an average of thirty independent trials.

In Table 1, performance results are shown for the BDRA using Methods 1 and 2, and a neural network. It can be seen in this table that the BDRA, using both methods, outperforms the neural network (a reference in [1] obtained an error of 0.115 with this data). Additionally, observe that the performance of Method 2 is superior to Method 1. This is because in Method 2 the fact that a feature is missing can represent relevant classification information. However, this is not the case with Method 1 where each missing feature is represented by a cardinality of possible values.

### 4 Summary

In this paper, the Bayesian Data Reduction Algorithm was extended for the case of missing features in either the training or test data. Overall, the algorithm was demonstrated to be superior to a neural network using both simulated and real data.

### References